REMARKS

The present response is intended to be fully responsive to all points of rejection raised by the Examiner and is believed to place the application in condition for allowance. Favorable reconsideration and allowance of the application is respectfully requested.

1. Claims

Applicants have amended claims 1, 3, 4, 6, 19, 20, and canceled claim 2 and 17. Now pending in this application are claims 1, 3-16, 18-20.

2. Response to Claim Rejections under 35 U.S.C. § 112

The Examiner has rejected claim 17 under 35 U.S.C. § 112, second paragraph, as being indefinite. In this response, Applicants have cancelled claim 17. Therefore, the Examiner's rejection is moot.

3. Response to Claim Rejections Under 35 U.S.C. § 102(e)

The Examiner has rejected claim 1-9, 11-13, 15, 16 and 18-20 under 35 U.S.C. § 102(e) as being anticipated by Walker et al. (US Patent No. 6,061,723).

Applicants have amended claims 1, 19 and 20 to more clearly distinguish the claimed invention from the teaching of Walker. In particular, claim 1 has been amended to include the feature of original claim 2, now cancelled. Amended claim 1 now recites that, for each event in the plurality of events, the method determines the number of devices and/or links between the device causing the event and the network management station when considering the location of the network device causing the event. In addition, claim 1 has been amended to recite that the causal event is determined as the event for which the determined number of devices and/or links is the fewest. This latter

feature is supported throughout the application, for example in the description at page 12, lines 1-3.

Claims 19 and 20 include corresponding amendments.

Applicants submit that Walker does not teach or suggest the method of the present invention as recited in amended claims 1, 19, and 20.

The technique disclosed in Walker is based on a criticalRoute attribute, which is defined column 6, line 28-32 as:

> "The criticalRoute attribute is a sequence of ovtopmd DB object identifiers which correspond to the route that a network packet could take if sent from netmon to a particular interface. The criticalRoute attribute traces the path of the intervening network interfaces."

In the network management system disclosed in Walker, called "netmon", the topology of a network is discovered, and during the first status poll of each node, the criticalRoute attribute is computed or validated on a per network interface basis. It is noted that Walker does not contain a description of how this criticalRoute attribute is calculated, and, in particular, how "the route that a network packet could take" is determined, although column 6, lines 49-56 suggests that is based on the route that a packet would take at that time. The method then polls the network interfaces in an order as displayed in the event browser, which indicates whether each interface is up, down or its status is unknown (see column 6, lines 22-27).

The method then uses an algorithm to determine which of the "interface down" events is a primary failure, i.e. the causal failure, and which other "interface down" events are due to a secondary failure (i.e. resulting from the primary failure). This algorithm is described from column 7, line 22 to column 8, line 48. This requires, for an "interface down" event, an examination of the status of every interface along the

criticalRoute to the interface that is down (column 7, lines 47-53), and, if no other interface is down, additional interrogation of the interfaces along the criticalRoute of an interface in question (IIQ) in order to determine whether the IIQ is down due to a primary failure (column 7, lines 62-67).

The Examiner takes the view that, in determining the primary failure, Walker considers the location of each network device/interface causing an event because of its use of the criticalRoute attribute, which traces the path between the network management system and the device/interface, and lists the intervening network interfaces including the interface in question, as illustrated in the table at column 6, lines 35-48. The Examiner also takes the view that Walker determines as the causal event, the event which has occurred at a location closest to the network management station.

Applicants would point out that the algorithm of Walker does not, strictly speaking, perform the steps of "considering the location of the network device causing each event" because the criticalRoute attribute does not define the relative locations of the network interfaces, but merely lists the interfaces, including the interface in question, in one possible path. The location of the interface in question is not considered by the algorithm. Instead the status of each interface along the critical route is determined by analyzing the stored status and, if necessary, polling each device to determine its status. If the status of a device along the criticalRoute is "down", then the event is a secondary failure. Otherwise, the event is a primary failure.

Moreover, Walker does not perform a count of the number of devices and/or links between the device causing the event and the network management station. Walker does

not determine or use the number of devices and/or links between the interface in question (IIQ) and netmon at all.

The Examiner indicates in the Office Action, in relation to claim 4, that in Walker "netmon can automatically determine the number of interfaces" in the critical path, since "the topology is known". Whilst the data available to netmon appears to be sufficient for such a count to be performed, there is no teaching or suggestion that netmon does this. On the contrary, it is evident that Walker does not count or otherwise determine the number of devices and/or links. Instead the method uses the criticalRoute attribute, which is simply a list of the identities (ovtopmd DB object identifiers) of the network interfaces in one possible path between netmon and the IIQ. Netmon then considers the status of each interface in the path defined of the criticalRoute attribute, one by one, by polling if necessary, until it finds an interface down (see column 8, lines 18-23). The number of interfaces in the criticalRoute attribute list is irrelevant to this process. The algorithm simple looks at each object identifier in the criticalRoute attribute in turn, and checks its current status (by polling, if necessary), and continues until it reaches the interface in question (IIQ). This is considerably more complex than simply counting the number of devices/links in the critical path, and involves the use of more processing and network resource.

The present invention, as defined in independent claims 1, 19 and 20, utilizes a simple and effective technique for determining the causal event in an event list by counting the number of devices and/or links in the path to the network management station for each event, and selecting the causal event as the event for which this number is smallest. Since this technique is neither taught nor suggested in the prior art, it is

submitted that the present invention as recited in each of independent claims 1, 19 and 20, is new and, furthermore, would not have been obvious to the skilled person at the date of the present invention.

Moreover, the dependent claims define further features of a preferred implementation of the present invention, which features are neither disclosed by, nor obvious in view of Walker. Therefore, the dependent claims 3-16, 18 which depend on independent claim 1 of the present application are novel as well.

For example, claim 7 defines the additional steps of determining whether a plurality of events are related before storing them in an event list. It is from this event list that the method of the present invention (as defined in claim 1) is carried out. This enables the technique to be used in the context of the different types of events contemplated in the present application. For example, the present invention applies to "high utilization on link" events and "slow responding service" events, as well as "link down" events.

The disclosure of Walker, in contrast, is merely concerned with handling "interface up/down" events, i.e. events describing devices or interfaces that are not responding at all. Such events result from "broken network elements" and "inoperative network elements".

Furthermore, the Examiner's objection to claim 7 does not appear to have appreciated that the events are not placed in the event list unless they are related, and, thus, the process of claim 1 is not carried out until it is determined that the events are related. The technique in Walker relies on the fact that all events, i.e. the status of all interfaces in the criticalRoute, are considered when determining the primary failure. The

method of claim 7 in contrast, only processes events that have been detected and have been determined to be related. These related events may not involve all devices in a given critical path.

4. Conclusion

Applicants submit that claims 1, 3-16, 18-20 are patentable over the prior art references cited by the Examiner. In light of the arguments set forth above, Applicants earnestly believe that the above claims are allowable. Therefore, Applicants respectively request that the Examiner expedite prosecution of this patent application to issuance.

Respectfully submitted,
McDONNELL BOEHNEN
HULBERT & BERGHOFF

Date: February 5, 2004

By: _____
Neilesh R. Patel
Reg. No. 50,918